TRW 130

AN/UYK-1 DATA PROCESSING SYSTEM

COMPUTER CONTROLLED CHECKOUT

APPLICATION STUDY

TRW COMPUTER DIVISION
THOMPSON RAMO WOOLDRIDGE INC

TRW 130

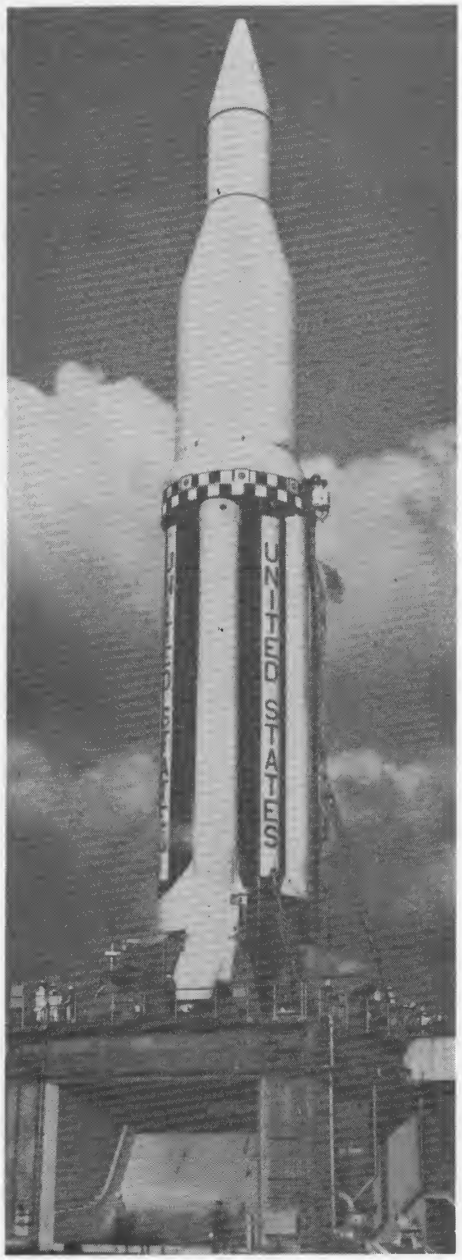
AN/UYK-1 DATA PROCESSING SYSTEM

COMPUTER CONTROLLED CHECKOUT

APPLICATION STUDY

TRW COMPUTER DIVISION
THOMPSON RAMO WOOLDRIDGE INC.
8433 Fallbrook Avenue • Canoga Park, California

M250-2U43 REV OCT 63


UNITED STATES

SECTION I

GENERAL TEST

SYSTEM REQUIREMENTS

Exacting methods required to insure the operational integrity of today's advanced aircraft and missile systems have established the need for mechanized checkout equipment. Possible equipment configuration can vary from rudimentary manually controlled systems to sophisticated stored program digital computer-controlled systems.

The specification of the test system is influenced by such factors as:

> prime system complexity
>
> maintenance capability at the test site, and
>
> operational requirements.

Obvious benefits accrue from using a single test system for an entire weapon system, rather than providing a special purpose test set for each electro-mechanical subsystem. A digital computer capable of real-time control is the ideal device for achieving this multiple-purpose capability. Further, the inclusion of a stored program digital computer in the test system makes it possible to perform certain data processing operations that are ancillary to the actual testing, but still necessary in the overall weapon system maintenance program. Some of these data processing functions are:

> statistical analysis of failure data
>
> reliability studies
>
> inventory control for replacement parts
>
> test report generation

- reduction of test data

- preparation of test programs.

Although the exact requirements for a particular test system depend upon specific conditions, the functions and capabilities which must be performed in any test situation can be defined as follows:

- Switching — making connections to input and response points; selecting stimuli.

- Comparing — validation of test responses.

- Sequencing — selection and ordering of required tests.

- Monitoring — continuous monitoring of bi-stable responses.

- Test control — selection, execution and termination of a series of tests under automatic control.

- Emergency response — corrective action taken upon recognition of any emergency condition.

- Manual control — communication between the automatic checkout system and the test conductor.

- Diagnosis — isolation of a fault to the level at which corrective action can be taken.

- Self-checking — periodic automatic verification of the proper operation of the checkout system.

- Computation — performance of calculations ranging from simple calibration of data to simultaneous equations for system simulation.

- Input — insertion of the test program.

- Output — real-time readouts of test results.

- Storage — provision for verification and fault isolation programs and for storing test results.

Within the boundaries of the specific conditions in the test system it is possible to define the following criteria for a digital computer.

- High speed multiple input-output cables capable of transmitting data to and from external devices.

- Computation and logical operation speeds compatible for real-time data processing.

- Minimum precision for computation using 12 bit A/D inputs.

- Fast access storage for constants, operating programs and test results.

- Mass storage device for recording selected test results.

> Data processing capability to perform ancillary functions such as: post test data reduction, test program preparation, and system diagnostics.

SECTION II

TEST SYSTEM

PROGRAMS

The operations required of the digital computer in the test system will vary from system to system and will depend upon the functional division of responsibilities between special hardware devices, such as: comparators, sequencers, stimuli selectors, and the computer. The exact division of responsibility and scope of testing can only be defined for a particular system and testing environment. General types of computer programs, however, are common to any computer-controlled test system.

TEST PROCEDURE PROGRAMS

The actual test procedure requires a program to control sequences, test limit conditions, make decisions based on test results, select values for output, select stimuli generators, and display test progress to the test conductor.

TEST PROCESS CONTROL PROGRAM

The test process control program selects, in sequence, the appropriate test procedure program on the basis of program command, previous test results, or command of the test conductor. This program also records test values selected by the various test procedure programs.

PROGRAM PREPARATION

A program is required to prepare a computer program from English language statements which describe the desired test. Only by the incorporation of such a compiler can the computer be used efficiently in the test system.

POST-TEST PROCESSING PROGRAM

Programs are required to process the recorded test results into meaningful information in a useful format. These programs would perform normal tape search, arithmetic computation, logical formatting, merging, filing updating and conversion.

The TRW-130 Digital Computer has:

> three input-output cables
> eleven real-time interrupt lines
> two hardware levels of interrupts
> programmable interrupts
> input bit rate at 999 KC
> six microsecond memory cycle time
> fifteen bit word length
> 8 K - 32 K word core memory and arithmetic and logical commands.

The TRW-130 is:

> programmable in machine language or interpretive mode, and is:
> manufactured to military specifications
> shock and vibration resistant and
> operable without temperature or humidity control.

The TRW-130 computer system has the capacity for:

> sixteen magnetic tape units

> line printer

> card reader

> paper tape reader/punch

> typewriter and

> memory expandable to 32 K in 8 K increments.

SECTION IV

TEST SYSTEM SOFTWARE

From an operational standpoint, the test system will require two kinds of programs: normal data processing programs which are expected to remain relatively stable; and programs which are generated by test technicians for controlling the system in the conduct of specific tests. The latter programs are written as the need arises, and will be subject to frequent modification as the nature of the system under test changes or as the testing philosophy changes. It is essential, therefore, that the programming structure be organized so that the test procedures can be written rapidly and accurately; as well as to permit these procedures to be readily integrated into the system.

TEST PROCEDURE LANGUAGE

For purposes of preparing test procedures, a compiler system is proposed which will permit test technicians to write test statements in a restricted English language, which makes free use of words, phrases and symbols peculiar to testing activities. In general, test statements will consist of (a) an identification; (b) a verb in the imperative mood, commanding some action; (c) an object on which this action is to be carried out; and (d) one or more qualifiers to particularize the verb and/or the object. For example:

$$\overbrace{\text{AT 100 SECONDS}}^{\text{Identifier}} \qquad \overbrace{\text{MEASURE}}^{\text{Verb}}$$

$$\overbrace{\text{DC VOLTAGE AT}}^{\text{Object}} \qquad \overbrace{\text{J31}}^{\text{Qualifier}}$$

The result of such a statement is to have the computer initiate a test time, 100 seconds, a sequence of operations which will measure the DC voltage at test point J31, and leave the digitized value in a standard computer location where it can be referred to by subsequent statements. For example, the statement

$$\overbrace{\text{RECORD}}^{\text{Verb}} \qquad \overbrace{\text{AS EMITTER-VOLTS}}^{\text{Qualifier}}$$

would cause the just-measured value to be stored in symbolic memory location EMITTER-VOLTS, thus permitting reference to this value at any later time by giving the address EMITTER-VOLTS. The action taken by the computer for these statements closely parallels that of a technician reading a DC voltage from a voltmeter and recording it on a data sheet under a column headed EMITTER VOLTS.

Since the purpose of taking measurements is to validate the item being tested, conditional statements are required. Such statements, in general, consist of a conditional phrase followed by one or more imperative statements of the type referred to above. For example: the statement

> IF EMITTER-VOLTS IS LESS THAN 6.2, GO TO
> FAILURE ROUTINE

would cause the computer to compare the quantity at EMITTER-VOLTS to 6.2, and if less, proceed to a standard failure routine.

The three statements illustrated above are fairly common in testing, and might profitably be combined into the following single statement.

AT 100 SECONDS, MONITOR DC VOLTAGE
(GREATER THAN 6.2) AT J31.

Such a statement could be defined initially into the language; or it could be defined by the procedure writer as required. For the latter purpose, a DEFINE statement would be provided, allowing the procedure writer to define arbitrary verbs in terms of the standard ones:

```
DEFINE MONITOR X Y Z AS
     MEASURE X Z
     IF Y, PROCEED; OTHERWISE
     GO TO FAILURE ROUTINE
```

Having made this definition, the compiler would subsequently interpret MONITOR statements in terms of the stated definition:

EXAMPLE: the statement

MONITOR AC CURRENT (LESS THAN 10 TIMES
COLLECTOR-CURRENT) BETWEEN Z14 AND Z19

would be interpreted by the compiler as the group of statements

```
MEASURE AC CURRENT BETWEEN Z14 AND Z19
IF LESS THAN 10 TIMES COLLECTOR-CURRENT,
     PROCEED; OTHERWISE GO TO FAILURE ROUTINE.
```

In addition to calling for specific actions at specific times on time intervals during the test, the test procedure writer may wish to control the recording and printing activity of the test process. For this purpose, special statements may be provided, such as:

SELECT M17, M18, M21, M22, M23 FOR PRINTING

AND

SELECT J24, J25, J30 FOR RECORDING.

During the test, the selected information would be printed and recorded on magnetic tape.

Before test procedures can be run on the computer, they must be reduced to a form more suitable for computer interpretation. The level of the language to which test procedures are reduced can range from basic machine language to a language slightly less abstract than the test procedure language. The choice will be a compromise between computer efficiency (obtained at the lower levels), and ease of program modification and interpretation by humans (obtained at the higher levels).

In addition to basic machine language (LOGic commANDS—logands), the TRW-130 computer provides a convenient intermediate level programming language which, because it was planned in the design of the computer, achieves computer efficiency approaching that of the basic machine language. This is the logram (LOGic progRAM) level, in which "instructions" of arbitrary complexity are implemented by collections of machine-language instructions or logands. Lograms resemble closed subroutines. However, the design of the TRW-130 greatly facilitates the linkage between lograms and the accessing of

logram parameters, with the result that special "bookkeeping" operations normally required in subroutines are not necessary.

The format adopted for logram instructions consists of a mnemonic operation code, followed by an arbitrary number or parameters or symbolic parameter addresses. For example, to add two 30-bit fields ALPHA and BETA and store the result at GAMMA, the sequence

LD2/ALPHA
AD2/BETA
ST2/GAMMA

would be written, where LD2, AD2, and ST2 identify the double-precision load, add, and store lograms, respectively. When assembled, this sequence would be converted to a series of binary addresses representing the starting addresses, the lograms, and the addresses of the associated parameters.

Location	Contents
Y	(Address of LD2 logram)
Y+1	(Address of "ALPHA")
Y+2	(Address of ADZ logram)
Y+3	(Address "BETA")
Y+4	(Address of ST2 logram)
Y+5	(Address "GAMMA")

Lograms currently exist for such relatively simple operations as load, store, add, subtract, multiply, divide, and branch, as well as for more complicated operations such as square root, binary to BCD conversion, sine, and arctangent.

A program system has been developed for the TRW-130 which allows the programmer to intermix symbolic logram instructions and symbolic basic machine instructions in a single program. This provides great flexibility in the construction of programs. An assembly program exists for converting such programs to absolute form.

TEST ORIENTED LOGRAM SET

It is proposed that a logram instruction set having the general properties outlined above becomes the target language for the test statement compiler. In addition to the usual computer functions, this set would contain instructions specifically oriented to a particular test equipment. In general, the scope of the latter type of instruction will be commensurate with that of the commands which the computer issues to the test equipment. The instructions themselves will consist of a mnemonic operation code followed by one or more parameters to designate an item of equipment, programmable operations to be performed by that equipment, and any digital values required by the equipment. Examples are:

SET/ANST/DCV/-14.3

Set DC voltage in Analog Stimuli
Section to -14.3 volts.

SEL/SSW/K13

Connect Patch panel point K13 to
Analog Stimuli Section.

SEL/RSW/J12

Connect Patch panel point J12 to Measurements Section.

SET/MEAS/DCC/100

DC current meter in Measurement Section to 100 ma range and sample input from Response Switching.

When assembled, these instructions would appear in computer memory as a string of binary addresses or numbers:

Location	Contents
Y	(Address of SET logram)
Y+1	(Address of Analog Stimuli Section)
Y+2	(Address of DC Voltage Source)
Y+3	(-14.3)
Y+4	(Address of SEL logram)
Y+5	(Address of Response Switching Section)

etc.

The principal duties of the lograms corresponding to these instructions would consist of: Assembling the appropriate command word; issuing the command to an external device; testing the response of the external device; and, if appropriate, waiting for the command to be completed and issuing the appropriate reset commands.

These test oriented instructions are somewhat more abstract than machine-language, but at the same time are considerably more detailed

than test procedure statements. They reflect the static properties of the test equipment, as opposed to the sequence of operations this equipment performs in the conduct of a test.

Just as the professional TRW-130 programmer has two programming levels at his disposal, so will the test procedure writer be able to prepare statements in either the test procedure language, test oriented instructions, or a combination of each. Thus the procedure writer has a flexible means of preparing test programs using a special problem-oriented language coupled with a general-purpose computer language. An example of combining the two languages is as follows:

```
APPLY PRESSURE TO J-31
MEASURE PRESSURE AT J-33
RECORD AS PSI

LD1/PSI
MP1/PSI
MP1/X            PSI. X + 1.5 = Z
AD1/*1.5
ST1/Z
IF Z IS GREATER THAN 14, GO TO STEP 10
```

TEST PROCEDURE COMPILER

The purpose of the test procedure compiler is to accept as input restricted English language test statements and program control statements, and procedure a sequence of logram instructions required for execution of the test procedure. For example, the statements

```
MEASURE DC VOLTAGE AT J31
RECORD AS EMITTER-VOLTS
IF EMITTER-VOLTS IS LESS THAN 6.2, GO TO
    FAILURE ROUTINE
```

might produce the following logram instructions:

```
SEL/RSW/J31
SET/MEAS/DC10
READ/$INPUT
MV1/$INPUT/EMITT
LD1/EMITT
CL1/*6.2/FAIL
```

To take advantage of the existing assembly program, and to keep the compiling process as simple as possible, it is proposed that compilation be performed in two steps: (1) translation and (2) assembly.

In the translation step, the input statements are converted to symbolic logram instructions of the form indicated above. In this step, the input statements are verified for:

a. Compatibility of verbs, objects, and modifiers in the statements;

b. Correctness of data formats in statements; and

c. Consistency of the sequencing of test statements.

Errors in any of these areas are printed out as they occur, with operator option to halt or proceed with the translation.

The translation step produces as output either a magnetic tape or punched paper tape consisting of the original test statements and the equivalent symbolic instructions. This tape is input to the second step, assembly, during which symbolic instructions are converted to absolute form and the required lograms are called in from the logram library. Output of the assembly is either magnetic tape or punched paper tape containing the assembled program in blocks of a size suitable for subsequent execution, and a "side-by-side" listing of the original statements, the symbolic instructions, and the absolute instructions. The listings would appear as shown in Table I.

The assembly program for performing step two has been written for both the TRW-130, and the IBM-7090. From the standpoint of utilizing this assembly program, therefore, the translator program could be prepared equally well for either computer. Experience with the assembly program indicates that the 7090 assembles about twice as fast as the TRW-130 when comparable input/output procedures are used on both machines. Since the translation involves the same kind of internal processing as assembly, it is reasonable to assume that the 2-to-1 ratio would hold for translation as well.

The use of the TRW-130 to compile test procedures is thus technically feasible, and offers the additional advantages of user convenience and straightforward operating procedures.

CHECKOUT OF COMPILED TEST PROCEDURES

Prior to integrating a test program into the test procedure library, the program must be validated to insure that it will perform its intended function, and that it will not misuse the test equipment or the vehicle

Table I. Assembled Program Output

STEP	STATEMENT	BLOCK NO.	LOCATION	INSTRUCTION SYMBOLIC	INSTRUCTION ABSOLUTE
101	MEASURE DC VOLTAGE AT J31	04	100	SEL	01342
			101	RSW	00013
			102	J31	00723
			103	SET	01410
			104	MEAS	00007
			105	DCV	00013
			106	10	00002
			107	READ	01333
			108	INPUT	06323
102	RECORD AS EMITTER-VOLTS	04	109	MVI	01310
			110	INPUT	06323
			111	EMITT	06143
103	IF EMITTER-VOLTS IS LESS THAN 6.2, GO TO FAILURE ROUTINE	04	112	LD1	01323
			113	EMITT	06143
			114	CLI	01347
			115	6.2	00062
			116	FAILU	02473

under test. For this purpose, a test program checkout system is proposed which will allow the test engineer to initiate a "dry-run" of this program from the test conductor's console, and observe printouts on a typewriter or printer of the actions taken by the computer in executing his program. This information is displayed in a manner which is meaningful to the engineer, so that he can readily detect inconsistencies in his program and initiate the appropriate corrections.

In the checkout mode, the logram instructions comprising the test procedure are executed in the trace mode, so that the results of each instruction may be displayed. The trace mode is also used to intercept instructions which refer to the vehicle to be tested, so that commands normally issued to and results normally received from the vehicle may be suitably simulated. This simulation may be accomplished internally to the computer, or by using suitable reference signals in the test equipment to simulate signals from the vehicle under test.

RELATED DOCUMENTS

M250-2U16

Introduction to the TRW-130 Data Processing System

M250-2U3

Standard Peripheral Equipment Group

M250-1U19

Reliability Prediction

M250-2U4

Installation Data

M250-2U5

A Programmer's Guide